基于RASCIL的W-Projection和W-Stacking并行算法实测研究*

杨秋萍[1,2,3]， 朵琳[3]

1，中国科学院云南天文台，云南 昆明 650216
2，中国科学院大学，北京， 100049
3，昆明理工大学信息工程与自动化学院，云南 昆明，650500

摘 要：在射电干涉阵的大视场成像中，W-Projection和W-Stacking是两类主要的成像方法。本文对这两种成像方法进行了并行实测研究。首先分析了两种成像方法的基本原理框架，在此基础上对两种成像方法并行实现时的关键因素进行了讨论和分析。利用已经校准的射电干涉阵观测数据对两种成像方法基于RASCIL分别进行并行策略研究和并行计算实验。通过对并行计算时间、并行效率和并行资源配置模式的分析，得到了两种成像方法基于RASCIL的并行计算性能，结果表明两种成像方法都适合采用Strong Scaling的并行资源配置模式进行并行计算，基于RASCIL的W-Stacking并行计算还有比较大的性能提升空间。

关键词：射电干涉阵；大视场成像；W-Projection；W-Stacking；并行计算； RASCIL

中图分类号：P164　　**文献标识码：**A　　**文章编号：**

1 引 言

在射电干涉阵的成像研究中，随着观测视场的扩大，射电干涉阵成像时需要进行大视场成像。射电干涉阵的大视场成像方法包括三维傅里叶变换的方法[1][2]，Faceting方法[3]，W-Projection方法[3]，W-Stacking方法等[5][6]。

SKA平方公里阵的提出为射电干涉阵的成像带来了新的挑战。SKA的巨大规模和复杂程度远远超出了现有射电天文望远镜阵列，全规模运行的SKA产生的海量数据需要10亿亿次/秒处理能力，是2017年最快的超级计算机神威太湖之光处理能力（0.9亿亿次/秒）的10倍[7]，因此在射电干涉阵的数据处理中，多节点的并行化实现一直是研究的重要内容。

在射电干涉阵的成像中，大视场成像方法的并行实现已成为研究的重点。劳保强等[10]实现了W-Projection的CPU并行和GPU并行的实验，并在天河二号（MilkyWay-2）超级计算机上进行了实验，还比较了单精度和双精度情况下数据加载时间和网格化运行时间； 以及对uv域的Faceting成像方法的并行化进行了研究[8]。于昂等[9]在uv域的Faceting成像方法和W-Projection成像方法的基础上提出了一种新的方法w-facets，并通过多核CPU和GPU进行了并行实现。Barnett等[11]提出新的网格化核函数，并通过并行实现测试核函数在网格化时的性能。Arras等[12]对提出的网格网方法进行了并行测试，给出了Strong Scaling下测试结果。

本文主要基于RASCIL[1]对W-Projection和W-Stacking方法利用DASK进行CPU并行实现，分析在RASCIL中两种成像方法在并行时的策略选择，以及并行资源的消耗，为后续基于RASCIL的射电干涉阵数据处理提供参考。第二部分介绍两种成像方法的基本实现框架；第三部分介绍两种成像方法的并行计算实验；第四部分是实验结果及讨论；最后是全文的结论。

2 W-Projection和W-Stacking的实现框架

W-Projection和W-Stacking是射电干涉阵大视场成像的经典算法，W-Projection是CASA[2]处

---

[1] https://gitlab.com/ska-telescope/external/rascil
[2] https://casadocs.readthedocs.io/en/stable/

理大视场成像的方法之一，而W-Stacking是WSClean[3]处理大视场成像的方法之一。

图1是W-Projection成像方法实现的功能框图，在图中最左侧是射电干涉阵接收的可见度数据集，在成像前该可见度数据集已完成校准，数据集加载后，可以获得观测的(u,v,w)的分布，利用这些信息通过式（1）计算得到成像时所需要的w平面数N_w，其中$\delta_A=0.02$ [5]，FOV为成像的视场，w_{step}是相邻的w平面的间隔，w_{max}是w的最大值，

$$w_{step}=\frac{\sqrt{2\delta_A}}{\pi FOV^2}$$
$$N_w=int\left(\frac{1.1\times 2.0\times w_{max}}{w_{step}}\right) \qquad (1)$$

然后利用卷积核函数生成N_w个卷积核，这也是W-Projection成像方法中的关键，具有不同w值的可见度数据与对应的卷积核进行卷积，实现从(u,v,w)向(u,v)平面的投影，卷积完的可见度数据将分布在(u,v)的平面上，并且进行叠加，最后通过傅里叶逆变换得到成像的脏图。

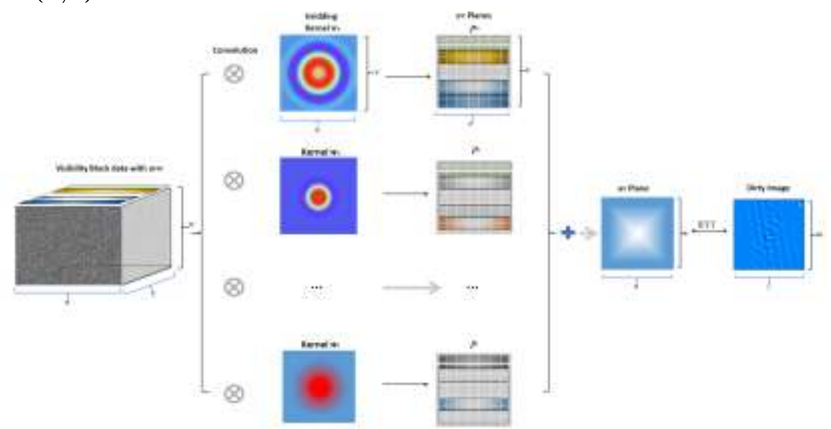


图1 W-Projection实现的原理框图

Figure 1 principles of W-Projection implementation

图2是W-Stacking成像方法实现的功能框图，在图中最左侧是射电干涉阵接收到的可见度数据集，同样该数据集是已经完成校准的。数据集加载后，与W-Projection相似，从(u,v,w)的分布通过式（1）计算出需要的w平面数N_w，把可见度数据集按照不同的w值分为N_w个数据切片，每一个数据切片都与卷积核进行卷积网格化，这个卷积核是分布在(u,v)平面的。卷积网格化后每一个数据切片通过傅里叶逆变换得到该数据切片所对应的图像，每一幅图像需要乘以$e^{-j2\pi w_{N_w}(\sqrt{1-l^2-m^2}-1)}$，然后再将所有图像进行相加合并，合并后再乘以$\frac{w_{max}-w_{min}}{\sqrt{1-l^2-m^2}}$，就得到W-Stacking成像的脏图。

[3] https://gitlab.com/aroffringa/wsclean/

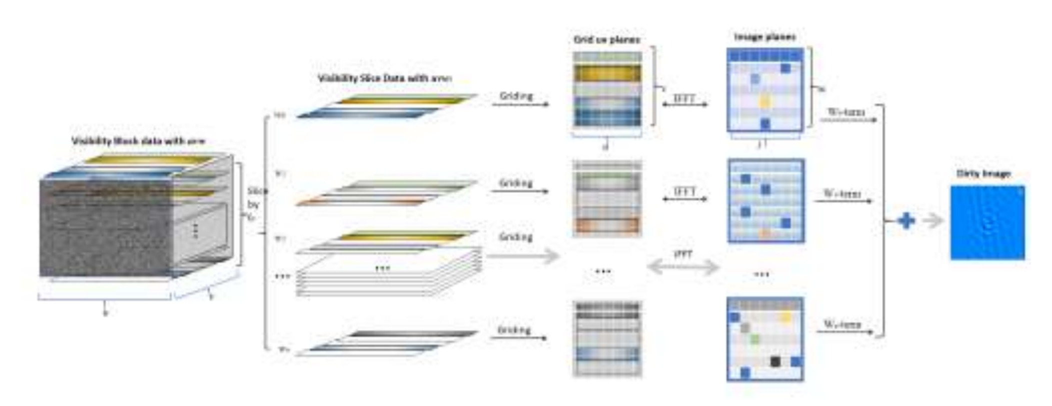


图2 W-Stacking实现的原理框图
Figure 2 principles of W-Stacking implementation

通过对两种成像方法实现框图的分析， W-Projectin方法与W-Stacking方法相同的之处是两种方法都需要根据观测校准的可见度数据的(u,v,w)的信息计算出w平面数N_w。而两种成像方法不同的是：W-Projection方法需要根据w平面数来计算投影的卷积核，卷积核的数量与N_w一致，在计算时需要考虑卷积核的宽度，以及过采样点的数量，这两个因素会影响卷积核的计算时间和在内存中占用的空间。 而W-Stacking需要根据w平面数，对可见度数据进行划分，需要将对应w值得可见度数据划分到同一个数据切片中，数据切片集合的数量与w平面数一致N_w。W-Stacking在卷积网格化这个环节中只需要一个卷积核，也就是所有数据切片都会与同样的卷积核进行卷积网格化，网格化后再通过傅里叶变换得到数据切片对应的图像。

在成像处理中W-Projection需要计算与N_w数量相等的卷积核，因此这一部分的计算消耗将超过W-Stacking，但是由于W-Stacking方法需要首先将可见度数据集进行划分数据切片，所以当可见度数据的规模达到一定时，需要特别考虑这一操作的计算效率；又因为W-Stacking在实现时需要存储与w平面数相等个数的数据切片，因此当可见度数据集具有一定规模时，需要的内存规模会远远超过W-Projection。

3 并行实验

3.1 实验数据

成像实验数据是来自Karl G.Janskey Very Large Array 阵列的D阵型观测校准数据SNR_G55_10s.calib.ms， 观测的目标是超新星遗迹G055.7+3.4，（观测的相位中心，RA：19:21:40，DEC:21.45.00），观测日期是2010-08-23-01:07:14.00（UTC）至2010-08-23-08:14:54.00（UTC），观测频段是L频段1-2GHz频率，包含4个频谱窗口，每个频谱窗口有64个频道，每个频道2种极化方式。数据的大小为1.4G。根据观测阵列的u,v,w分布情况，可以计算出最长基线和最大的w值，再根据成像时每个像素的大小为8角秒，图像每个方向包含1280个像素，就可以根据式（1）得出在上述成像参数下，改正w-term的影响所必需的w平面数为68。

3.2 并行计算

（1）并行环境

本文使用射电天文模拟、校准和成像库(RASCIL，Radio Astronomy Simulation, Calibration, and Imaging Library,)来进行成像。RASCIL是一个完全开源的射电干涉阵列数据处理包，已被广泛用于SKA数据模拟和处理研究。它的处理结果已经与其他软件进行了比较，并被证实是可靠的。RASCIL是用Python开发的，在实验中采用的版本为v.0.1.9。

并行处理基于DASK进行，DASK也是一个开源库，为现有的Python 堆栈提供并行性，DASK与Python库集成。在科学计算中数据集和计算规模的扩展速度远远高于处理器和内存发展的速

度，科学计算往往需要扩展到多台计算机进行运算，DASK提供了可跨多个核心、处理器和计算机实现并行执行。

（2）实验硬件配置

成像实验中使用相同配置的硬件作为计算节点，硬件配置如表1所示。各个节点通过光纤万兆交换机和路由器进行连接组成以太网，保证各计算节点在同一局域网组网，每个节点有相同的网络延迟和带宽。

表1 并行计算节点的硬件配置

Table 1 Hardware configuration of Parallel computing nodes

Hardware	Model	Specification	Quantity
Motherboard	Intel(R)	C610	1
CPU	Intel(R) Xeon(R) E5-2650L v4	14 cores 1.7Ghz 14 nm TDP65w	2
RAM	Samsung DDR4	DDR4 2400 CRC	192GB
Hard disk	Samsung SSD 960 EVO	512GB	2
Network Adapter	Intel X540-AT2	10G Ethernet	1

（3）实验软件配置

操作系统：Ubuntu 22.04 LTS，开发语言环境：Python v3.8.2，并行计算框架：DASK 2022.6.1，应用软件：RASCIL v0.1.9。

3.3 两种成像方法的并行策略

两种大视场成像方法的并行实现中，当处理器负载最大，或者数据流带宽达到上限时，计算系统的吞吐量达到上限饱和，计算系统将到达瓶颈，无法提升处理速度。因此，成像算法并行化设计的前提是在保证计算系统吞吐量最大时计算性能结果进行分析，即在不同算法和数据切片并行调度方式下，所分配的处理器资源或者带宽资源应接近100%占用率。

两种大视场成像算法的实现核心可以分成下面三个部分：

1）数据集加载和数据集的预处理；

2）计算任务和数据的分布式调度执行；

3）计算结果聚合，为下一环节pipeline做准备。

（1）W-Projection的并行策略

W-Projection能够分解成一些完全独立的子任务、同时各个子任务之间数据几乎没有依赖，没有通信。以每通道可见度数据分配并行计算任务，数据由vis_load_ms task读入，创建各通道图像和卷积核(create_wp_gcfcf_from_vis)，完成gridding相关invert和sum_invert task计算，并将各通道图像通过gahter_image将图像结果汇集，如图3所示，在图中需要持久化的计算环节用“P”做了标注。

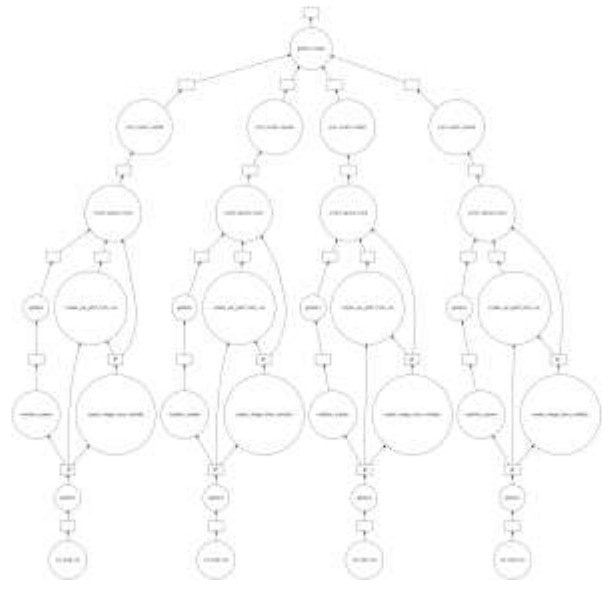


图3 W-Projection的并行策略

Figure 3 Parallel computing implementation of W-Projection

（2）W-Stacking的并行策略

W-Stacking以每通道可见度数据并行分配计算任务的方式，数据由load_ms task读入，每个通道做数据w_slice切片后进行 Scatter-Gather并行执行，分片（Scatter）之后每个分片进行gridding_invert task计算，计算结果进行聚合（Gahter），并将各通道图像结果汇集gahter_image。 W-Stacking按照w进行数据分片，这样有两种并行策略:

1）策略1 多通道并行，每通道分片并行计算，这样将会消耗非常多的内存资源， 任务并行关系示意图如图4左图，其中持久化的计算结果环节将用 “P”表示。由于RASCIL中数据分片数量和Task数量相等，这会导致在并行处理中系统内存开销过大，因此在目前的硬件配置下这种策略无法得到成功的计算结果。

2）策略2 多通道并行，每通道顺序分片并行计算，节省内存资源 任务并行关系示意图如图4右图，其中持久化的计算结果环节将用 P表示，因此在W-Stacking的并行实现时选择了这种策略。

在图3和图4中只画出了4条数据通道的任务图，在实验中可见度数据的通道数最高为64。

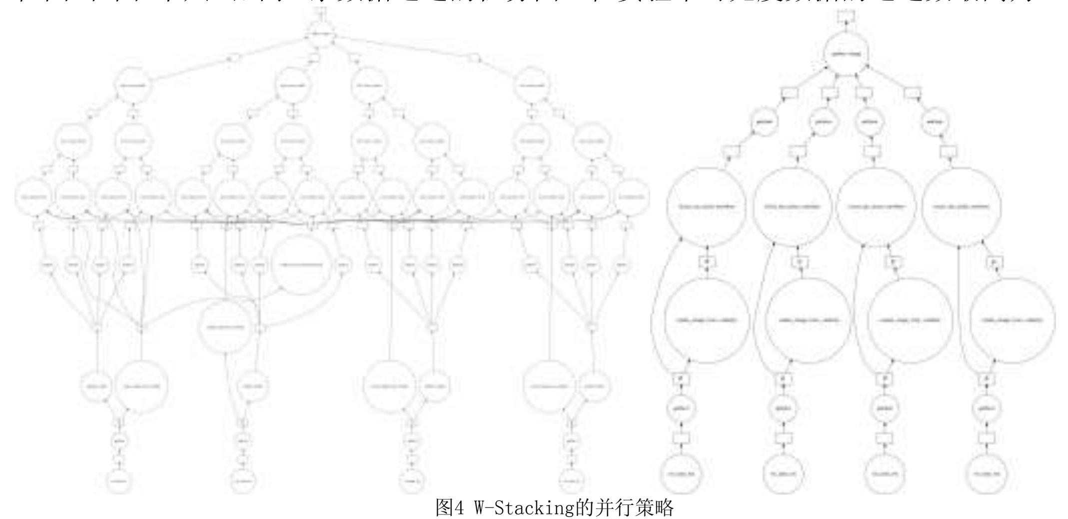


图4 W-Stacking的并行策略

Figure 4 Parallel computing implementation of W-Stacking

4 结果讨论

在成像方法的DASK并行计算实验中，为减少进程切换损耗和并行调度复杂度，每个worker对应一个进程和线程，worker总数量不超过处理器逻辑核心数量，每个worker最大内存限制为12GB。

在并行处理中最小参考基准是单一worker，单一物理核心，单一进程和线程，对于超线程架构和睿频技术的CPU核心，逻辑核心计算性能并不等同于物理核心计算性能。因此在并行计算时按照同一计算场景，同一节点上56逻辑核心和28物理核心计算时间（分别为208.49秒和190.94秒）进行如下的换算，

物理核心约等于$\frac{56/28}{208.49/190.94} \times \frac{2.5}{1.7} = 2.69$倍逻辑核心计算能力，其中单核心睿频2.5GHz，CPU的基频1.7GHz，两者的比值为处理器能力提升倍率。

4.1 并行计算时间

（1）在数据处理与计算规模恒定的情况下，即这时处理的可见度数据均为 64 通道的， 对应不同计算集群和CPU核心数量（每个 Worker 对应一个物理或逻辑核心）配置下， 对两种成像方法的并行计算完成的时间进行比较，如表2和表3所示，可以看出在增加节点或节点的 CPU

核心数量增加时， 两种成像方法的计算时间都降低。

（2）在集群节点数量都恒定为2，CPU物理核心数量随着处理数据规模等比例增加的情况下，并行计算完成的时间如表4所示，可以看到，当数据规模不断增加后，随着物理核心数的增加，计算时间也在增加。

表2 W-Projection并行计算时间（单位：秒）

Table 2 Parallel computing time of W-Projection (unit: second)

Number of nodes in cluster	Number of visibility channel	Physical cores per node	Equivalent logic cores	Execution time
1	64	1	2.69	1174
1	64	n/a	1	3158
1	64	14	28	208.49
1	64	28	56	190.94
2	64	8	43.06	145.363
2	64	14	75.36	122.657
2	64	56	112	98.87

表3 W-Stacking并行计算时间（单位：秒）

Table 3 Parallel computing time of W-Stacking (unit: second)

Number of nodes in cluster	Number of visibility channels	Physical cores per node	Equivalent logic cores	Execution time
1	64	1	2.69	3843
1	64	n/a	1	10327
1	64	14	28	512.61
1	64	28	56	580.80
2	64	8	43.06	429.251
2	64	14	75.36	324.400
2	64	56	112	260.70

表4 数据规模和计算核心同步增长时并行计算时间（单位：秒）

Table 4 Parallel computing time for simultaneous expansion of data size and computing cores (unit: second)

Number of visibility channels	Physical cores	W-Projection execution time	W-Stacking execution time
1	1	12.617	54.984
2	2	21.647	60.139
4	4	20.306	63.185
8	8	35.238	69.301
16	16	37.992	94.277
28	28	61.054	134.960
36	36	88.371	191.821
48	48	105.620	266.238

4.2 成像方法并行效率分析

在并行处理中，加速比的定义如式（2）所示，其中在上式中S_p为加速比，E_p是并行效率，T_s为最优顺序算法的单处理核心执行时间，T_p为使用p个处理器核心并行计算所花费的时间。

$$S_p = \frac{T_s}{T_p}$$
$$E_p = \frac{S_p}{p} \times 100\% = \frac{T_s}{p * T_p} \quad (2)$$

并行加速比和两种成像方法并行效率如图5所示，其中左图为两种成像方法的并行加速比，右图为并行效率。由于并行任务的分配并不一定和worker数量整除尽，计算时间和系统并行效率并不是随运行CPU核心数量（Worker数量）线性增加和减少的关系，所以图中出现核心数多反

而更慢的起伏点。处理密集运算（多核心计算）时候，并行效率衰减较快，这是因为一方面并行调度开销的增长，另外一方面现代CPU睿频和超线程和多CPU的架构特性为少量核心运算的计算场景提供更好每核心性能。

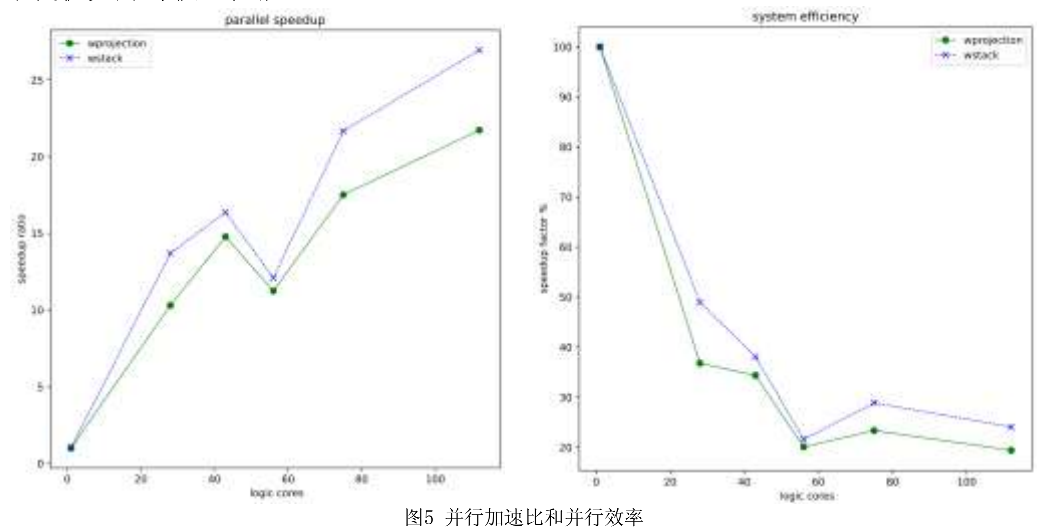


图5 并行加速比和并行效率
Figure 5 Parallel speedup and system efficiency

4.3 数据规模分析

当处理的可见度数据规模发生变化时，针对两种并行计算资源配置的不同场景进行分析：

（1）Strong Scaling：在并行处理中，问题规模保持不变(即可见度数据的通道数量不变），增加处理器数量，这用于找到解该问题最合适的处理器数量，即所用时间尽可能短而又不产生太大的开销。两种成像方法在Strong Scaling资源配置模式的并行计算时间与加速比如图6，左图为W-Projection，右图为W-Stacking，从图中可以看出，在处理的可见度数据通道数量不变时，W-Projection在43核心时达到计算速度与每核心使用效率的最佳平衡点，而W-Stacking在28核时达到。

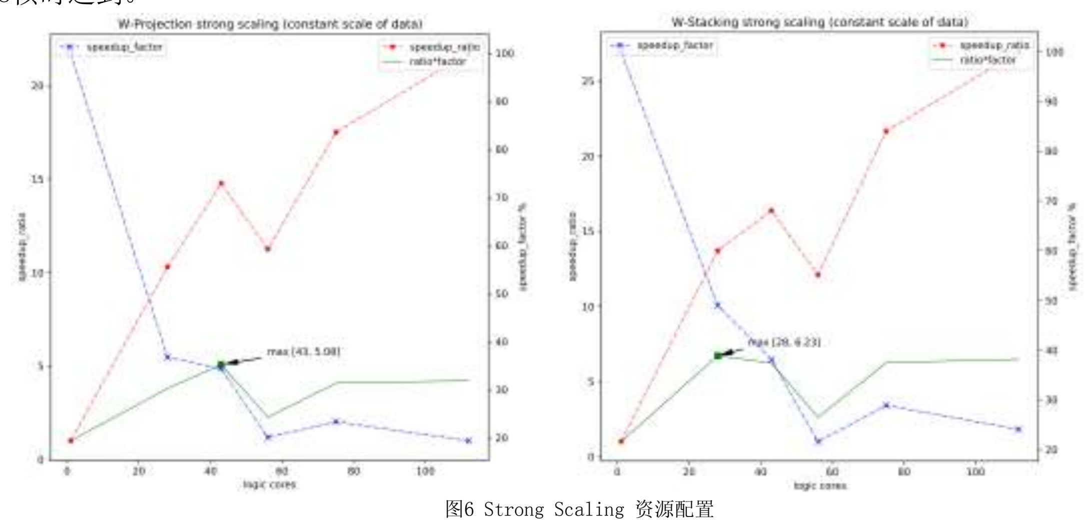


图6 Strong Scaling 资源配置
Figure 6 Strong Scaling: parallel computing configuration

（2）Weak Scaling：并行处理问题规模（计算量）随处理器数量增加而增加，适合这种并行资源配置策略时并行效率会保持水平稳定。测试结果如图7所示，从图中可以看出，并行效率迅速下降，因此两种成像方法的并行实现在数据规模发生变化时并不适合Weak Scaling并行资源配置策略。

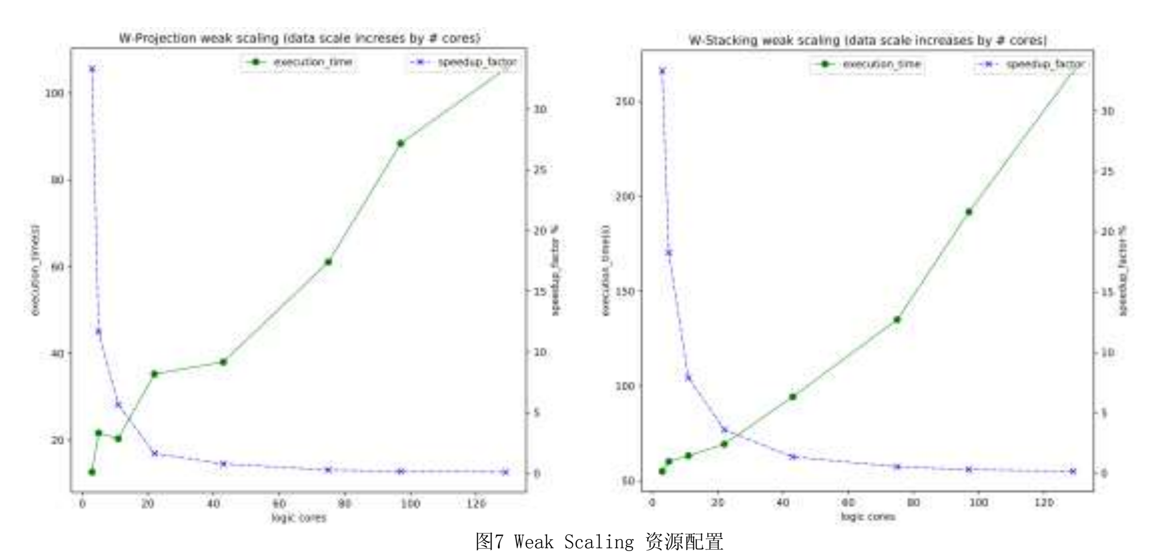


图7 Weak Scaling 资源配置

Figure 7 Weak Scaling: parallel computing configuration

5 结论

本文首先介绍了W-Projection和W-Stacking两种大视场成像方法的实现原理框图，在此基础上完成基于RASCIL的W-Projection和W-Stacking方法的DASK并行实验，采用了VLA-D阵列对超新星遗迹G055.7+3.4的校准观测数据集进行上述两种成像方法的并行处理，对两种成像方法设计了并行计算策略，在不同集群节点和处理器核心数（即worker数量）配置下，分别在数据规模恒定和数据规模随处理器增长两种场景做了并行计算时间统计，对加速比和并行实现效率进行分析，并在数据规模增长时对比了两种成像方法在Strong Scaling 和 Weak Scaling两种策略下并行计算的优劣,两种成像方法适合采用Strong Scaling并行资源配置方式。由于在RASCIL v0.1.9版本中，可见度数据是利用visibility对象进行存储和访问的，在进行并行计算时该对象变得非常庞大，而W-Stacking方法需要多次对visibility对象进行读写，所以该方法在并行计算时消耗的时间超过了W-Projection，在未来基于RASCIL的成像并行计算中，可以考虑对可见度数据采用更加简单的数据结构，这样就能大幅提升W-Stacking的成像效率。

An Empirical Study of W-Projection and W-Stacking Parallel Computing Based on RASCIL

Yang QiuPing[1,2,3], Duo Lin[3]

(1, Yunnan Observatories，Yunnan， Kunming, 650216, China
2, University of Chinese Academy of Sciences, Beijing, 100049, China
3, Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500, China)

Abstract: W-Projection and W-Stacking are classical wide field imaging methods in radio interferometry. In this paper, the parallel computing implementation of these two imaging methods is carried out. First, the basic principle of the two imaging methods is analyzed, based on which the key factors for the parallel computing implementation of the imaging methods are discussed. The parallel computing experiments are carried out based on RASCIL for these imaging methods using calibrated radio interferometry observation data. The performances of the two imaging methods parallel computing are obtained by analyzing the parallel computation time, as well as the parallel efficiency and parallel resource configuration. The results show that both imaging methods are suitable for the parallel computing configuration of Strong Scaling, while the parallel computing of W-Stacking based on RASCIL has a great potential for performance improvement.

Key words: radio interferometry; wide field imaging; W-Projection; W-Stacking; parallel computing; RASCIL

参考文献:

[1] Sramek R, Schwab F R. Imaging [C]//Synthesis Imaging in Radio Astronomy: volume 6. 1989: 117-138.

[2] Cornwell T J, Perley R A. Radio-interferometric imaging of very large fields [J]. Astronomy and Astrophysics, 1992, 261: 353-364.

[3] Perley R A. Synthesis imaging in radio astronomy [C]//Astronomical Society of the Pacific Conference Series: volume 6. 1989.

[4] Cornwell T J, Golap K, Bhatnagar S. Astronomical data analysis software and systems xiv [C]// Astronomical Society of the Pacific Conference Series: volume 347. 2005.

[5] Humphreys B, Cornwell T J. Ska memo 132 [C]//Square Kilometre Array Memo: volume 132. 2011.

[6] Offringa A R, McKinley B, Hurley-Walker N, et al. wsclean: an implementation of a fast, generic wide-field imager for radio astronomy [J/OL]. Monthly Notices of the Royal Astronomical Society, 2014, 444: 606-619. DOI: 10.1093/mnras/stu1368.

[7] 戴伟, 汪森, 李秋虹, 邓辉, 梅盈, 王锋. 基于 Spark 的 SKA1-MID 自校准管线分布计算实现 [J]. 天文研究与技术, 2020, 17: 334-34

Dai Wei, Wang Sen, Li Qiuhong, Deng Hui, Mei Ying, Wang Feng. Implementation of SKA1-MID Self-calibrating Pipeline Based on Spark. Astronomical Research and Technology, 2020, 17(3): 334-340.

[8] 劳保强, 安涛, 于昂, 等. uv-faceting 成像并行算法研究 [J/OL]. 天文学报, 2019, 60. DOI: 10.15940/j.cnki.0001-5245.2019.02.012.

LAO Bao-qiang, An Tao, Yu Ang, et al. Research on Parallel Algorithms for uv-faceting Imaging[J/OL]. Acta Astronomica Sinica, 2019, 60. DOI: 10.15940/j.cnki.0001-5245.2019.02.012.

[9] 于昂, 劳保强, 王俊义, 等. 混合 w-facets 成像并行算法研究 [J/OL]. 天文学进展, 2020, 38. DOI: 10.3969/j.issn.1000-8349.2020.04.05.

YU Ang, LAO Bao-qiang, WANG Jun-yi, AN Tao. Research on Parallel Algorithms for Hybrid w-facets Imaging[J/OL]. Prograss in Astronomy,2020,38. DOI: 10.3969/j.issn.1000-8349.2020.04.05.

[10] Lao B Q, An T, Yu A, et al. Parallel implementation of w-projection wide-field imaging [J/OL]. Science Bulletin, 2019, 64: 586-594. DOI: 10.1016/j.scib.2019.04.004.

[11] Barnett A H, Magland J, af Klinteberg L. A parallel nonuniform fast fourier transform library based on an“exponential of semicircle” kernel [J/OL]. SIAM Journal on Scientific Computing, 2019, 41(5): C479-C504. DOI: 10.1137/18M120885X.

[12] Arras P, Reinecke M, Westermann R, et al. Efficient wide-field radio interferometry response [J/OL]. Astronomy & Astrophysics, 2021, 646: A58. DOI: 10.1051/0004-6361/202039723.

收稿日期：；　修订日期：

*基金项目：　太阳低波段地空图像的配准方法研究，国家自然基金面上项目，11773012，2018 年 1 月-2021 年 12 月

作者简介：杨秋萍(1979-)，女，河北宁晋人，主要从射电干涉阵数据处理方面的研究工作。